

Multivariate probit regression using simulated maximum likelihood

Lorenzo Cappellari & Stephen P. Jenkins

ISER, University of Essex

stephenj@essex.ac.uk

Overview

- Introduction and motivation
- The model and the method of Simulated Maximum Likelihood (SML)
- The **mvprobit** program (**ml**, method **lf**)
- Illustrations – **mvprobit** in action
- Further remarks about program use

Introduction and motivation

- Evaluation of probit model likelihood functions requires calculation of Normal probability distribution functions.
- Algorithms exist for accurately calculating accurate univariate and bivariate Normal pdfs, but not for trivariate or higher dimensional Normal distributions (at least not in Stata). Instead, ...
- Recent literature on calculating multivariate Normal pdfs using simulation-based methods
- Here: multivariate probit model estimated using simulated ML ('GHK' simulator): **mvprobit**
 - cf. **triprobit** at SSC-IDEAS

The model

M equation multivariate probit model:

$$y_{im}^* = \beta_m' X_{im} + \varepsilon_{im}, m = 1, \dots, M$$

$$y_{im} = 1 \text{ if } y_{im}^* > 0 \text{ and } 0 \text{ otherwise}$$

ε_{im} , $m = 1, \dots, M$, are error terms distributed as multivariate normal, each with a mean of zero, and variance-covariance matrix V , where V has values of 1 on the leading diagonal and correlations $\rho_{jk} = \rho_{kj}$ as off-diagonal elements.

- Structure like a SUR model but depvars are binary (and need not have same set of X in every equation)
- M different choices at a point in time OR choices on one item at M points in time (panel model with free correlations)

Estimation principles ($M = 3$)

Log-likelihood function for a sample of N independent observations:

$$L = \sum_i w_i \log \Phi_3(\mu_i; \Omega)$$

w_i is an optional weight

$\Phi_3(\mu_i; \Omega)$ is standard trivariate normal cdf, where

$$\mu_i = (K_{i1}\beta_1'X_{i1}, K_{i2}\beta_2'X_{i2}, K_{i3}\beta_3'X_{i3})$$

with $K_{ik} = 2y_{ik} - 1$, for each $j, k = 1, \dots, 3$.

Ω has elements Ω_{jk} , where $\Omega_{jj} = 1$ for $j = 1, \dots, 3$; $\Omega_{21} = \Omega_{12} = K_{i1}K_{i2}\rho_{21}$, $\Omega_{31} = \Omega_{13} = K_{i3}K_{i1}\rho_{31}$, $\Omega_{32} = \Omega_{23} = K_{i3}K_{i2}\rho_{32}$.

- Log-likelihood function depends on the trivariate standard normal distribution function $\Phi_3(\cdot)$!
- Evaluate using Geweke-Hajivassiliou-Keane (GHK) smooth recursive conditioning simulator

The GHK simulator

- Exploits the fact that a multivariate normal distribution function can be expressed as the product of sequentially conditioned univariate normal distribution functions – which can be easily and accurately evaluated.
- Trivariate case: 8 joint probabilities corresponding to the eight possible combinations of successes ($y_{im} = 1$) and failures ($y_{im} = 0$). Focus on $\Pr(\text{every outcome is a success})$:

$$\begin{aligned}\Pr(y_1 = 1, y_2 = 1, y_3 = 1) &= \Pr(\epsilon_1 \leq \beta_1'X_1, \epsilon_2 \leq \beta_2'X_2, \epsilon_3 \leq \beta_3'X_3) \\ &= \Pr(\epsilon_3 \leq \beta_3'X_3 \mid \epsilon_2 < \beta_2'X_2, \epsilon_1 < \beta_1'X_1) \\ &\quad \times \Pr(\epsilon_2 < \beta_2'X_2 \mid \epsilon_1 < \beta_1'X_1) \times \Pr(\epsilon_1 < \beta_1'X_1)\end{aligned}$$

- Expression involves conditioning upon unobservable variables (that are correlated with each other).
- However if a good approximation for these conditional distributions can be found, then the likelihood function only requires evaluation of univariate integrals.
- How may the approximations be derived?

The GHK simulator (ctd.)

Cholesky decomposition of the covariance matrix for the errors:

$$E(\varepsilon\varepsilon') \equiv V = Cee'C$$

where C is the lower triangular Cholesky matrix corresponding to V and $e \sim \Phi_3(0, I_3)$, i.e. three uncorrelated standard normal variates.

Hence:

$$\varepsilon_1 = C_{11}e_1$$

$$\varepsilon_2 = C_{11}e_1 + C_{22}e_2$$

$$\varepsilon_3 = C_{31}e_1 + C_{32}e_2 + C_{33}e_3$$

and C_{jk} is the jk th element of matrix C .

\Rightarrow rewrite $\Pr(\text{three successes})$ as

$$\Pr(\varepsilon_1 \leq \beta_1'X_1, \varepsilon_2 \leq \beta_2'X_2, \varepsilon_3 \leq \beta_3'X_3)$$

$$= \Pr[e_3 \leq (\beta_3'X_3 - C_{32}e_2 - C_{31}e_1)/C_{33} \mid e_2 \leq (\beta_2'X_2 - C_{21}e_1)/C_{22}, e_1 \leq \beta_1'X_1/C_{11}] \\ \times \Pr[e_2 \leq (\beta_2'X_2 - C_{21}e_1)/C_{22} \mid e_1 \leq \beta_1'X_1/C_{11}] \times \Pr[e_1 \leq \beta_1'X_1/C_{11}].$$

The standard normal variates, e , that now appear in the decomposition are uncorrelated with each other (by construction).

The first two conditional probabilities can be further rewritten as unconditional probabilities defined in terms of truncated standard normal variates:

The GHK simulator (ctd.)

Rewriting in terms of truncated standard normal variates:

$$\begin{aligned} & \Pr(\epsilon_1 \leq \beta_1'X_1, \epsilon_2 \leq \beta_2'X_2, \epsilon_3 \leq \beta_3'X_3) \\ &= \Pr[\epsilon_3 \leq (\beta_3'X_3 - C_{32}e_2^* - C_{31}e_1^*)/C_{33}] \times \Pr[\epsilon_2 \leq (\beta_2'X_2 - C_{21}e_1^*)/C_{22}] \\ & \quad \times \Pr[\epsilon_1 \leq \beta_1'X_1/C_{11}] \\ &= Q_3 \times Q_2 \times Q_1, \text{ say,} \end{aligned}$$

where e_1^* and e_2^* are truncated univariate standard normal variates with upper truncation points at $\beta_1'X_1/C_{11}$ and $(\beta_2'X_2 - C_{21}e_1^*)/C_{22}$ respectively.

Computation of Q_1 is straightforward and, if one had some specific values for e_1^* and e_2^* , then one could also compute Q_2 and Q_3 , and hence the overall multivariate probability.

Similar arguments for all the other probabilities of success/failure (and for $M > 3$).

The GHK simulator (ctd.)

GHK simulator:

- (i) Derive values for e_1^* , e_2^* via random draws from upper truncated standard normal distributions with truncation points as above (use random number generator, plus inversion formula);
- (ii) Recursively compute a multivariate probability value from the Q s.
- (iii) Replicate R times, and then
- (iv) Calculate the simulated probability as the arithmetic mean of the values of the simulated probabilities from each replication.

SML estimator: ML with the multivariate normal probabilities calculated at each iteration using the GHK simulator \Rightarrow numerically intensive!

- SML estimator consistent as $N \rightarrow \infty$ and $R \rightarrow \infty$. (Asymptotic, like ML!)
- Simulation bias $\rightarrow 0$ when R raised with N . [$(R / \sqrt{N}) \rightarrow \infty$ sufficient.]

The **mvprobit** program

```
mvprobit equation1 equation2 ... equationM  
    [weight] [if exp] [in range] [, draws(#) seed(#)  
    beta0 atrho0(matrix_name) robust cluster(varname)  
    constraints(numlist) level(#) maximize_options ]
```

where each equation is specified as

```
( [eqname:] depvar [=] [varlist] [, noconstant] )
```

by may be used; all types of weights allowed; has standard features of estimation commands including access to estimated results; no limit on M (in principle)

Options

draws(#) : number of random draws in simulation (R). Default = 5.

seed(#) : initial value of random-number seed used in simulation process.
(Default = 123456789).

beta0: estimates of the marginal probit regressions are reported.

atrho0(*matrix_name*) : starting values for the off-diagonal elements of the correlation matrix V that differ from the default starting values (all zero).

Remaining options: same as the corresponding ones for **biprobit**.

Prediction using **mvppred**

mvppred newvarname_prefix [if exp] [in range] [, statistic]

where **statistic** is one of

xb the linear prediction for each equation; the default.

stdp the standard error of the linear predictions for each equation.

pmarg the marginal success probability for each equation.

pull the joint probabilities: (i) $\Pr(y_{im} = 1, \text{ for all } m = 1, \dots, M)$, and
(ii) $\Pr(y_{im} = 0, \text{ for all } m = 1, \dots, M)$.

Only one statistic may be chosen at a time.

For statistics **xb**, **stdp**, and **pmarg**, results are stored in the variables **newvarname_prefix*i***, for equations $i = 1, \dots, M$.

For the **pull** statistics, results are stored in the variables **newvarname_prefix1s** for predicted probability (i) and **newvarname_prefix0s** for predicted probability (ii).

[Options for prediction restricted to the ‘all successes’ and ‘all failures’ cases = the only two cases that could be programmed without M being fixed. (Number of joint probabilities is 2^M .)]

Illustrations

(1) (i) syntax, options etc., and (ii) accuracy of SML relative to ML (**mvprobit** vs. **biprobit**)

– ‘School’ data (Pyndyck & Rubinfeld; Stata manuals), $N = 95$, $M = 2$

(2) Simulated data, $N = 5000$, $M = 4$.

‘School’ data:

`private`: whether children attend a private school,

`vote`: whether the household head had voted for an increase in the property tax,

`years`: # years family has been at the present residence

`logptax`: $\log(\text{property tax})$; `loginc`: $\log(\text{income})$.

biprobit estimates:

```
. use http://www.stata-press.com/data/r7/school.dta, clear

. biprobit (private=years logptax loginc) (vote=years logptax loginc), nolog
```

Seemingly unrelated bivariate probit

Number of obs	=	95
Wald chi2(6)	=	9.59
Prob > chi2	=	0.1431

Log likelihood = -89.254028

	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
-----+-----						
private						
years	-.0118884	.0256778	-0.46	0.643	-.0622159	.0384391
logptax	-.1066962	.6669782	-0.16	0.873	-1.413949	1.200557
loginc	.3762037	.5306484	0.71	0.478	-.663848	1.416255
_cons	-4.184694	4.837817	-0.86	0.387	-13.66664	5.297253
-----+-----						
vote						
years	-.0168561	.0147834	-1.14	0.254	-.0458309	.0121188
logptax	-1.288707	.5752266	-2.24	0.025	-2.416131	-.1612839
loginc	.998286	.4403565	2.27	0.023	.1352031	1.861369
_cons	-.5360573	4.068509	-0.13	0.895	-8.510188	7.438073
-----+-----						
/athrho	-.2764525	.2412099	-1.15	0.252	-.7492153	.1963102
-----+-----						
rho	-.2696186	.2236753			-.6346806	.1938267
-----+-----						

Likelihood ratio test of rho=0: chi2(1) = 1.38444 Prob > chi2 = 0.2393

```
. predict p11,p11
. predict p00,p00
. predict xbb1, xb1
. predict xbb2, xb2
. predict stdpb1, stdp1
. predict stdpb2, stdp2
. predict pmargb1, pmarg1
. predict pmargb2, pmarg2
```

Predictions of joint & marginal probabilities,
linear predictions, etc.

mvprobit estimates ($R = 100$) are close to ML estimates:

```
. mvprobit (private = years logptax loginc) (vote=years logptax loginc), nolog
> dr(100)
```

```
Multivariate probit (SML, # draws = 100)      Number of obs   =          95
                                                Wald chi2(6)     =          9.64
Log likelihood = -89.220805                    Prob > chi2      =          0.1405
```

	Coef.	Std. Err.	z	P> z	[95% Conf. Interval]	
-----+-----						
private						
years	-.0118233	.0256205	-0.46	0.644	-.0620386	.0383921
logptax	-.1033056	.6672673	-0.15	0.877	-1.411126	1.204514
loginc	.3695001	.5303571	0.70	0.486	-.6699806	1.408981
_cons	-4.140149	4.837923	-0.86	0.392	-13.6223	5.342006
-----+-----						
vote						
years	-.0172153	.0148029	-1.16	0.245	-.0462285	.011798
logptax	-1.280732	.5725493	-2.24	0.025	-2.402908	-.1585563
loginc	.9956743	.437904	2.27	0.023	.1373982	1.85395
_cons	-.5627991	4.055359	-0.14	0.890	-8.511157	7.385558
-----+-----						
/atrho21	-.2811165	.2396506	-1.17	0.241	-.7508231	.18859
-----+-----						
rho21	-.2739382	.2216667	-1.24	0.217	-.6356397	.1863855
-----+-----						

```
Likelihood ratio test of rho21 = 0:
      chi2(1) = 1.45088    Prob > chi2 = 0.2284
```

```
. mvppred pall, pall
(Pr(all zeros), Pr(all ones) will be stored in variables pall0s, pall1s)
. mvppred xbm, xbm
(xbm will be stored in variables xbm1, i = 1,...,#eqs)
. mvppred stdpm, stdpm
(stdpm will be stored in variables stdpm1, i = 1,...,#eqs)
. mvppred pmargm, pmargm
(pmargm will be stored in variables pmargm1, i = 1,...,#eqs)
```

Predictions

SML predictions very similar to ML counterparts

```
. su pall1s p11 pall0s p00 xbm1 xbb1 xbm2 xbb2 /*
>          */ stdpm1 stdpb1 stdpm2 stdpb2 pmargm1 pmargb1 pmargm2 pmargb2
```

Variable	Obs	Mean	Std. Dev.	Min	Max
pall1s	95	.0513848	.0293697	.0006823	.1675037
p11	95	.0514965	.0295284	.0006783	.1691212
pall0s	95	.3252403	.1496049	.0397381	.8772917
p00	95	.3241522	.1485598	.040815	.882799
xbm1	95	-1.273431	.2017621	-1.930628	-.8744448
xbb1	95	-1.275218	.2041617	-1.937996	-.8695227
xbm2	95	.3308476	.4381363	-1.365069	1.519954
xbb2	95	.3313479	.4383708	-1.37215	1.52224
stdpm1	95	.3404043	.1805338	.185824	1.046698
stdpb1	95	.3405953	.1807651	.1859334	1.049172
stdpm2	95	.2541217	.1181758	.141525	.7976922
stdpb2	95	.2546185	.1186652	.1415696	.8023056
pmargm1	95	.1057509	.0322633	.0267645	.190938
pmargb1	95	.1055308	.032576	.0263119	.1922807
pmargm2	95	.6216642	.1502225	.0861157	.9357387
pmargb2	95	.6218135	.1500823	.0850083	.9360256

Less similarity between SML and ML with smaller R , we found.

More on choice of R below.

$N = 95$ is ‘small’, so potential finite sample biases anyway.

Raise N and M in simulated data example ...

Simulated data: $M = 4$, $N = 5000$

```
. set seed 12309
. set obs 5000
obs was 0, now 5000

. matrix R = (1, .25, .5, .75 \ .25, 1, .75, .5 \ .5, .75, 1, .75 \ .75, .5, .7
> 5, 1)

. drawnorm u1 u2 u3 u4, corr(R)

. corr u*
(obs=5000)
```

Correlation structure (V)

	u1	u2	u3	u4
u1	1.0000			
u2	0.2587	1.0000		
u3	0.5077	0.7483	1.0000	
u4	0.7523	0.5093	0.7589	1.0000

```
. ge x1 = uniform()-.5
. ge x2 = uniform() + 1/3
. ge x3 = 2*uniform() + .5
. ge x4 = .5*uniform() - 1/3
```

```
. * Equations
. ge y1s = .5 + 4*x1 + u1
. ge y2s = 3 + .5*x1 - 3*x2 + u2
. ge y3s = 1 - 2*x1 + .4*x2 -.75*x3 + u3
. ge y4s = -6 + 1*x1 - .3*x2 + 3*x3 - .4*x4 + u4
.
. ge y1 = y1s>0
. ge y2 = y2s>0
. ge y3 = y3s>0
. ge y4 = y4s>0
```

Equations

mvprobit estimates ($R = 75$)

Estimates quite close
to those in the ‘true’
model

NB warning message at
iteration 1 not a problem
as model later converged

LR test of MVP against M
independent univariate
probits (V identity matrix)

```
. mvprobit (y1=x1) (y2=x1 x2) (y3 = x1 x2 x3) (y4=x1 x2 x3 x4), dr(75)
```

Iteration 0: log likelihood = -8681.8526
Warning: cannot do Cholesky factorization of rho matrix
Iteration 1: log likelihood = -7922.4199
Iteration 2: log likelihood = -7749.5212
Iteration 3: log likelihood = -7746.5769
Iteration 4: log likelihood = -7746.5734
Iteration 5: log likelihood = -7746.5734

Multivariate probit (SML, # draws = 75) Number of obs = 5000
Wald chi2(10) = 5561.10
Log likelihood = -7746.5734 Prob > chi2 = 0.0000

		Coef.	Std. Err.	z	P> z	[95% Conf. Interval]
y1	x1	3.991634	.0962586	41.47	0.000	3.80297 4.180297
	_cons	.5078066	.0233257	21.77	0.000	.462089 .5535241
y2	x1	.5413448	.0704179	7.69	0.000	.4033283 .6793614
	x2	-2.848419	.0781794	-36.43	0.000	-3.001648 -2.695191
	_cons	2.867846	.0734467	39.05	0.000	2.723894 3.011799
y3	x1	-2.011164	.0708565	-28.38	0.000	-2.15004 -1.872288
	x2	.5341271	.0642228	8.32	0.000	.4082527 .6600015
	x3	-.7438451	.0311735	-23.86	0.000	-.8049441 -.6827462
	_cons	.9133876	.0725944	12.58	0.000	.7711051 1.05567
y4	x1	1.125271	.0891551	12.62	0.000	.9505303 1.300012
	x2	-.3030878	.0826195	-3.67	0.000	-.4650191 -.1411565
	x3	2.898115	.0779738	37.17	0.000	2.745289 3.050941
	x4	-.4352364	.1598866	-2.72	0.006	-.7486084 -.1218644
	_cons	-5.751499	.1685475	-34.12	0.000	-6.081846 -5.421152
/atrho21		.2621885	.0317031	8.27	0.000	.2000516 .3243253
/atrho31		.5646645	.0345949	16.32	0.000	.4968597 .6324692
/atrho41		.9396437	.0571792	16.43	0.000	.8275746 1.051713
/atrho32		.9737444	.0406156	23.97	0.000	.8941392 1.05335
/atrho42		.5670195	.0424459	13.36	0.000	.4838271 .6502119
/atrho43		1.007126	.0537097	18.75	0.000	.9018571 1.112395
rho21		.2563413	.0296198	8.65	0.000	.1974249 .3134127
rho31		.5114301	.0255462	20.02	0.000	.4596439 .5597501
rho41		.7350585	.0262846	27.97	0.000	.6791715 .7824714
rho32		.7503451	.0177483	42.28	0.000	.7134322 .7831052
rho42		.513167	.0312682	16.41	0.000	.4493034 .5718126
rho43		.7645708	.0223127	34.27	0.000	.7172009 .8049075

Likelihood ratio test of rho21 = rho31 = rho41 = rho32 = rho42 = rho43 = 0:
chi2(6) = 1870.56 Prob > chi2 = 0.0000

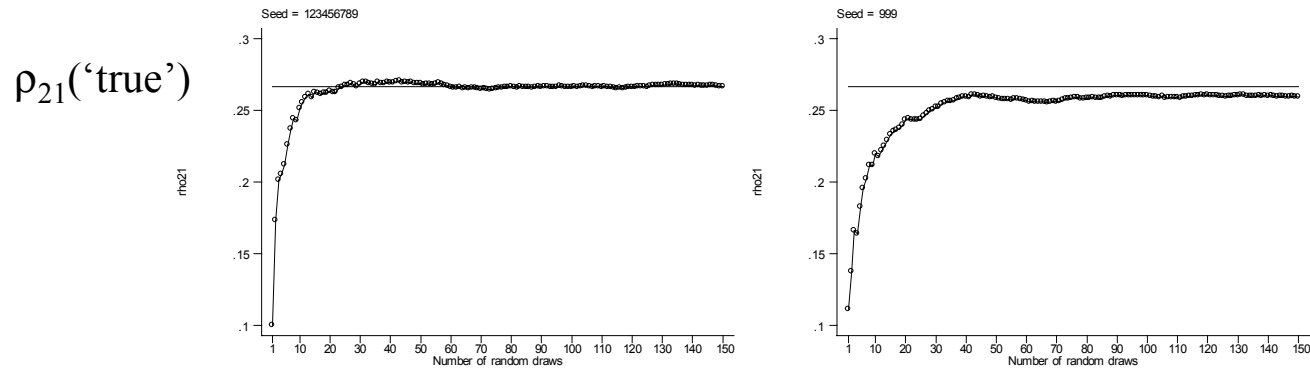
Using `mvprobit`: further remarks

- Choosing number of random draws: higher R increases accuracy but also run-time;
- Choosing seed (different random numbers give different simulated probabilities)

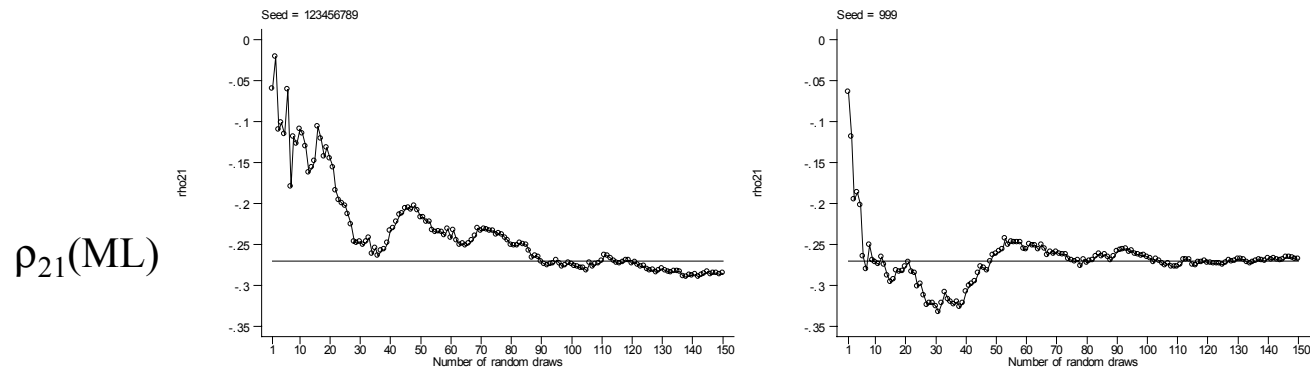
Re-estimated the earlier models for each value of $R = 1, \dots, 150$, and several alternative seed values

SML estimate of ρ_{21} : $R = 1, \dots, 150$; seed = 123456789, 999

(a) two-equation model, generated data ($N = 5000$)



(a) two-equation model, 'School' data ($N = 95$)



With 'large' N , SML estimator OK regardless of seed when $R > \sqrt{N}$

Using **mvprobit**: further remarks

Run-time:

- Four-equation model took c. 2.25 hours using Stata 7/SE on Pentium P4/1.4Ghz; c. 5.3 hours using Stata 7/IC on Sun Solaris)
- increases linearly in R (for given N, M)
- increasing number of explanatory variables, *ceteris paribus*, has no big effect
- **atrho** option use reduced run-time relatively little (since got good estimates of V within a few iterations anyway)
- Runtime increases substantially as M increases (several days/weeks if $N = 1000$ s and $M > 6$ or $7!$)

Capacity:

- $R*M$ temporary variables created: **set memory**
- For 'large' models (many covariates, many M): **set matsize**